Network Security

2021/2022

Solutions of the exercises

1) Let us consider a simple monoalphabetic shift cipher (Caesar's Cipher), with an alphabet of di N characters (with N= 26), with a secret key K=4 (the shift). Do encrypt the text "SECRET".

SOLUTION
A shift of *K=4* leads to the following encryption substitution table:
Cleartext char: `abcdefghijklmnopqrstuvwxyz`
Ciphertext char: `EFGHIJKLMNOPQRSTUVWXYZABCD`
then:
Cipher text: $c = E_k(m)$ = SHIFT(4,"SECRET") = "WIGVIX"

2) Consider a monoalphabetic substitution cipher, that maps a plaintext character *M* into the cipher character *C,* defined as follows:
$C = E_k(M) = a\ M + b \bmod 26$
where *M* is any character of the alphabet {'a','b', 'c', .. ,'z'}, and a and b are two integer parameters that form the secret key $K = \langle a,b \rangle$
By using such a cipher, a ciphertext has been generated starting from an English plaintext. By analyzing the ciphertext it results that the most frequent letter of the ciphertext is 'B', and the second most frequent letter of the ciphertext is 'U'.
Try to break this code, by knowing that the two most frequent letters in English are 'e' and 't'.
(Hints: $x \bmod n = y \Rightarrow \exists\ h : x = y + hn$. The equation $15x \bmod 26 = 19$ has the solution $x = 3$).

SOLUTION
M1='e'=4 → C1='B'=1
M2='t'=19 → C1='U'=20

$(4a+b) \bmod 26 = 1$
$(19a+b) \bmod 26 = 20$

$\exists\ h : b = 1-4a +h*n$
$(19a -4a +1 +h*26) \bmod 26 = 20$

$15a \bmod 26 = 19$, that has the solution $a=3$ (to find the solution, you can use the Euclid's algorithm to find 15^{-1} modulo 26, that is 7; then by multiplying both sides by 7 you obtain $a = 7*19 \bmod 26 = 3$)

Then, $a = 3$, and $b = 1 – 4*3 +h*26 = 15$

$C = E_k(M) = 3\ M + 15 \bmod 26$

3) Starting from a block cipher $E_K(\cdot)$ with block size *q*, please show the scheme for the CBC (Cipher Block Chaining) encryption of a message *m* with length *L>q* (for simplicity, let's consider *L=n q*).

SOLUTION
If we express *m* and *c* as:
$m=M1||M2|| \ldots ||M_n$

$c = C1 || C2 || \ldots || C_n$

it is:
$C_0 = IV$
$C_i = E_K(M_i \oplus C_{i-1})$

4) Suppose to have an API implementing a block cipher *E* in CBC mode, with block size *q*. The same block cipher in CBC mode has been used to encrypt a message *m* with length *pq* using a key *K* of size *n* bits. Evaluate the complexity of a brute force attack against the secret key *K*, by supposing to know both the plaintext *m* and the ciphertext *c*. In each attempt, the entire message is processed. Indicate the complexity in terms of the number of block encryptions (using the function *E*), as function of *n*, *p* and *q*.

SOLUTION
Given the message *m*, the maximum number of keys that should be tried (worst case) in order to find the right key *K* such that *E-CBC(K,m) ≡ c* is 2^n. Since each attempt requires the execution of *p* encryption operations, the complexity of this attack in terms of number of *E* operations is:
$\mathbf{p\ 2^n}$.
If T_E is the time for one encryption with E(·), the total time required for the complete brute-force attack is: $p\ 2^n\ T_E$
The same result could be obtained by using the decryption function *D-CBC(K,c)* and searching the key *K* such that: *D-CBC(K,c) ≡ m.*

5) Let us consider a symmetric block cipher $E_k(\cdot)$ with size 4 bit.
By supposing that, given a secret key *K*, the encryption table of $E_k(\cdot)$ corresponds to the table at the right side, do encrypt in CBC mode with IV=0000 the following plaintext message:
m= `1100 1010 0010 1101`

plaintext	***ciphertext***
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

SOLUTION

Encryption is performed in CBC mode, that is:
$C_i = E_k(M_i \text{ XOR } C_{i-1})$
with $C_0 = IV = 0000$

then:
$C_1 = E_k(1100 \text{ XOR } 0000) = E_k(1100) = 0101$
$C_2 = E_k(1010 \text{ XOR } 0101) = E_k(1111) = 0111$
$C_3 = E_k(0010 \text{ XOR } 0111) = E_k(0101) = 1111$
$C_4 = E_k(1101 \text{ XOR } 1111) = E_k(0010) = 1101$

c= `0101 0111 1111 1101` (iv=0000)

6) Let us consider the following plaintext message:
m = `1100 0000 1100 0000`
encrypted by means of the same symmetric encryption algorithm $E_k(\cdot)$ with block size 4bit and secret key *K* of the previous exercise (same encryption/substitution table) in OFB mode with IV=0001, resulting the following ciphertext:
c = `1000 0010 0001 1001 (IV=0001)`
Show how it is possible to modify the ciphertext c in such a way that by decrypting it you obtain the following plaintext:

m'= `1100 0000 1001 0000`

SOLUTION

Encryption has been done in OFB mode, that is c = m XOR o.
Hence, if you want to change a bit of the decrypted plaintext you have to change the corresponding bit of the ciphertext.
Referring to the third block:
original M_3 = 1100
target M_3' = 1001

so you have to simply change the second and fourth bit of C_3, that is:
original C_3 = 0001
modified C_3' = 0100

c'= `1000 0010 0100 1001 (iv=0001)`

7) Let us consider a message *m=M1||M2||M3||M4*, and suppose to decrypt it by means of a block cipher $E_K()$ in CBC mode (the block size of $E_K()$ is equal to the size of the blocks *Mi*), with *iv=IV0*, obtaining the ciphertext *c= C1||C2||C3||C4*. If an attacker modifies the ciphertext by rearranging the component blocks obtaining the new ciphertext c'= *C1||C3||C2||C4*, which will be the corresponding plaintext message *m'=M'1||M'2||M'3||M'4* obtained by "erroneously" decrypting the ciphertext *c'*? Show the blocks *M'i* as function of *Mj* and *Cj* with *j=1..4*.

SOLUTION
With CBC encryption, it is:
$C_i = E_K(M_i \oplus C_{i-1})$
and:
$M_i = D_K(C_i) \oplus C_{i-1}$

and also:
$D_K(C_i) = M_i \oplus C_{i-1}$

indicating with:
m' = M'1||M'2||M'3||M'4

by setting:
c'= *C1||C3||C2||C4*

it results:
$M'1 = D_K(C'1) \oplus IV0 = D_K(C1) \oplus IV0 = M1$
$M'2 = D_K(C'2) \oplus C'1 = D_K(C3) \oplus C1 = (M3 \oplus C2) \oplus C1$
$M'3 = D_K(C'3) \oplus C'2 = D_K(C2) \oplus C3 = (M2 \oplus C1) \oplus C3$
$M'4 = D_K(C'4) \oplus C'3 = D_K(C4) \oplus C2 = (M4 \oplus C3) \oplus C2$

8) Realize a symmetric encryption scheme for encrypting messages *m* with any length, based on a block cipher $E_K()$ (e.g. AES), without obtaining avalanche effect, in such a way that if you change one bit of the ciphertext, only one bit of the plaintext will change when decrypting the ciphertext (hint: use the XOR operator).

SOLUTION
$m=M1||M2|| \dots ||M_n$
$c=IV||C1||C2|| \dots ||C_n$
$C_i=M_i \oplus O_i$
with:
$O_i= E_K(O_{i-1})=AES(K,O_{i-1})$
$O_0=IV$

9) Consider the following three padding algorithms for extending the length of a message to a multiple of N bytes (e.g. N=32). Which of the three algorithms are suitable for using with a block cipher with block size N bytes? Why?

Padding1: append to the message random bytes until the total length (in bytes) becomes a multiple of N.

Padding2: append to the message random bytes until the total length (in bytes) becomes a multiple on N – 1; append one byte encoding the number of padding bytes that have been added.

Padding3: append to the message a bit ‘1’, then append as many bits ‘0’ as needed to reach a multiple of N bytes.

SOLUTION

All three padding algorithms extend the message length to a multiple of N. However only Padding2 and Padding3 are suitable for encryption/decryption, since they allow the receiver to detect the end of the original message and to correctly remove the padding data after decryption.

10) Starting from a hash function H() and a symmetric key K_{AB} shared between two entities A e B,

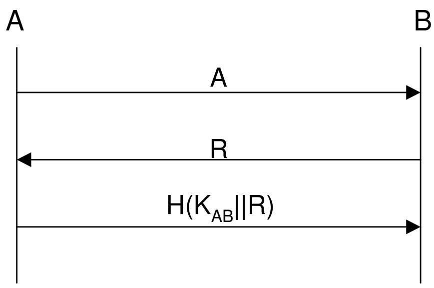
i) show a possible authentication scheme between A (supplicant) and B (authenticator)

ii) show how it is possible to send a message *m* from A to B providing data authentication and integrity protection

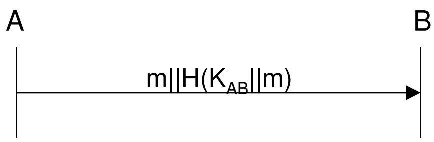
iii) create an encryption function (and the corresponding decryption function) that can be used for sending a message *m* encrypted from A to B

SOLUTION

i) A possible authentication scheme between A (supplicant) and B (authenticator):



ii) Authentication and integrity protection of a message *m* sent from A to B:



iii) Encryption function (and the corresponding decryption function) that can be used for encrypting the message *m* from A to B:

Let’s define

$O_0 = IV$

$O_i = H(K_{AB} \| O_{i-1})$

$o = O_1 \| O_2 \| O_3 \| \,..\, \| O_n \| \,..$

$c = E(K_{AB}, IV, m) = m \oplus o$

message that is sent:

$A \rightarrow B : x = IV \| c$

decryption:

$m = E(K_{AB}, IV, c) = c \oplus o$

11) Find the multiplicative inverse of each nonzero element in Z_7.

SOLUTION

$Z_7^* = \{1,2,3,4,5,6\}$

Corresponding multiplicative inverses: 1, 4, 5, 2, 3, 6

12) Find all nonzero elements in Z_{21} that are relatively prime with 21.

SOLUTION
Elements in Z_{21} that are co-prime with 21 are: $U_{21} = \{1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20\}$.
Note that: $\Phi(21) = \Phi(3x7) = (3-1)(7-1) = 12 = |U_{21}|$

13) By using the Euclid's algorithm, find the greatest common divisor gcd(,) of:
   a) 36, 15
   b) 47, 20
   c) 43, 35

SOLUTION
a) gcd(36,15)=(36,15)=(15,6)=(6,3)=3
b) gcd(47,20)=(20,7)=(7,6)=(6,1)=1
c) gcd(43,35)=(35,8)=(8,3)=(3,2) =(2,1)=1

14) Using Fermat's theorem, find 3^{201} mod 11.

SOLUTION
201 = 200 +1 = 20x10 +1 = 20x(11-1)+1.
Since p=11 is prime and gcd(3,11)=1, from the Fermat's theorem:
3^{10} mod 11 = 1
then:
3^{201} mod 11 = $3x(3^{10})^{20}$ mod 11 = $3x1^{20}$ = 3

15) Prove the following: If *p* and *q* are prime, then $\Phi(pq) = (p-1)(q-1)$.
   (Hint: What numbers have a factor in common with *pq*?)

SOLUTION
The integers that are less than *pq* and have a factor in common with *pq* are: p,2p,3p, .. (q-1)p, q,2q,3q, .. (p-1)p
In total they are (q-1) + (p-1) values.
Since the total number of values less than *pq* is: pq-1,
then:
$\Phi(pq) = pq-1 – [(q-1) + (p-1)] = pq –p –q +1 = (p-1)(q-1)$

16) Find λ , $\mu \in \mathbb{Z}$ such as $25 \lambda + 32 \mu = 1$, by using the extended Euclid's algorithm; use the result values for solving the equation $25 x \equiv 4 \mod 32$

SOLUTION
From the Extended Euclid's algorithm:
$r_k = a_k \cdot 32 + b_k \cdot 25$

with:
$r_k=r_{k-2} – r_{k-1}$
$a_k=a_{k-2} – a_{k-1}$
$b_k=b_{k-2} – b_{k-1}$

Starting from:
$32=1 \cdot 32 + 0 \cdot 25$
$25=0 \cdot 32 + 1 \cdot 25$

we have (Extended Euclid's algorithm):

rk	ak	bk
32	1	0
25	0	1
7	1	-1
4	-3	4
3	4	-5
1	-7	9

that resuts: $\lambda=9$ e $\mu=-7$, that leads to: $9 \cdot 25 - 7 \cdot 32 = 1$
obtaining:
$9 \cdot 25 = 1 - \mu \cdot 32$
that is:
$9 \cdot 25 = 1 \bmod 32$

The previous result can be used to solve the equation *25x=4 mod 32*, that is:
$25x=4 \bmod 32$
$x=25^{-1} \cdot 4 \bmod 32$
$x = 9 \cdot 4 \bmod 32 = 36 \bmod 32 = 4$

17) Create a pair of public/private RSA keys, using as *p* and *q* primes the values p=3, q=11. With such keys, do encrypt the plaintext message m=2.

SOLUTION
$n=pq=33$
$\phi(n)=(p-1)(q-1)=20$
Possible values for *e* and *d* are: 1,3,7,9,11,13,17,19 (co-primes of 20)
If we choose *e*=7
using the extended Euclid's algorithm:

20	1	0
7	0	1
6	1	-2
1	-1	3

that gives *d= 3*, with $ed=1 \bmod \phi(n)$
If we define the public and private keys as: $K^+=<e,n>$ and $K^-=<d,n>$
By encrypting m with the public key K^+ we have:
$c=E(m)=2^7 \bmod 33=29$

it is also possible to verify that:
$m=D(c)=29^3 \bmod 33=((29*29) \bmod 33)*29 \bmod 33)=16x29 \bmod 33=2$

18) With the following values p=7, q=11 and e=13. Create a pair of public/private RSA keys KU=<e,n> and KR=<d,n> (Use the Euclid's algorithm for finding the value d). With such keys, do decrypt the ciphertext message c=2.

SOLUTION
$n=77$, $\Phi(n)=60$
$e=13$

By using the extended Euclid's algorithm:

rk	ak	bk
60	1	0
13	0	1
8	1	-4
5	-1	5
3	2	-9
2	-3	14
1	5	-23

that leads to:
$1=5 \cdot 60-23 \cdot 13$
that is:
$(-23) \cdot 13 = \bmod 60$
$d=e^{-1}=(-23)=37$

Then:
$m=2^{37} \bmod 77=51$

Verify:
$51^{13} \bmod 77=2=c$

19) In a public-key system using RSA, you intercept the ciphertext $C = 10$ sent to a user whose public key is $e = 5$, $n = 35$.What is the plaintext M?

SOLUTION
It easy to find that n = 35 = 5x7, than p=5, q=7, $\Phi(pq)=24$.
Since it is: $d \cdot e = 1 \bmod 24$,
than d=5, and $M=10^d \bmod n = 10^5 \bmod 35 = 5$

20) In an RSA system, the public key of a given user is $e = 31$, $n = 901$.What is the private key of this user? (*Hint:* First use trial-and-error to determine p and q; then use the extended Euclidean algorithm to find d)

SOLUTION
By trying to divide n=901 by different prime p values, we find p=17, and q= n/p = 53.
Hence, $\Phi(n) = 16x52 = 832$, and (by using the Euclid's algorithm) $d= e^{-1} \bmod 832 = 671$.

21) Show an example of shared key exchange between A and B based on Diffie-Hellman scheme, using the generator g=2 and the prime p=11.

SOLUTION
If A chooses the secret $x_a=5$, while B chooses the secret $x_b=3$, we have (Diffie-Hellman exchange):
A send to B: $ya=g^{xa} \bmod p=10$
B send to A: $yb=g^{xb} \bmod p=8$
starting from ya and xb, B computes: $K_{BA} = ya^{xb} \bmod p=10^3=100x10=1x10=10$
starting from yb and xa, A computes: $K_{AB} = yb^{xa} \bmod p=8^5=(8^2)^2x8=9^2x8=4x8=10$
with: $K_{AB} = K_{BA}$

22) Show that 2 is a primitive root of 11.

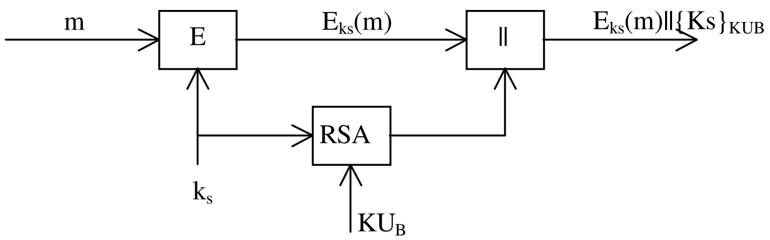
SOLUTION
By computing $g^1, g^2, .. g^k \bmod 11$, with g=2, we obtain: 2,4,8,5,10,9,7,3,6,1, that are all nonzero elements in Z_{11} that are co-prime with 11 (since 11 is prime, all nonzero integer less than 11 are coprime with 11); that means that 2 is a primitive root of 11.
Alternatively:
From the previous computed values, it is possible to see that the first m such as $g^m = 1 \bmod 11$, is $m=10=\Phi(11)$.

23) Users A and B use the Diffie-Hellman key exchange technique with a common prime p=71 and a primitive root g=7.
i. If user A has private key x_A=5, what is A's public key y_A?
ii. If user B has private key x_B=12, what is B's public key y_B?
iii. What is the shared secret key K_{AB}?

SOLUTION
$y_A = 7^5 \bmod 71 = 51$
$y_B = 7^{12} \bmod 71 = 4$
$K_{AB} = 4^5 \bmod 71 = 30 = 51^{12} \bmod 71 = 30$

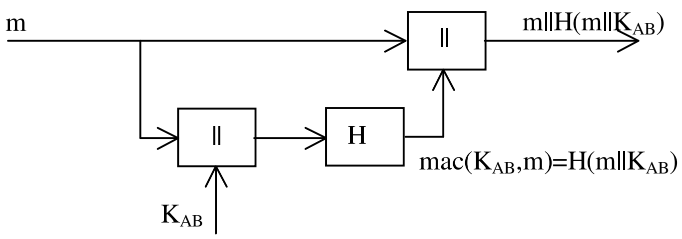
24) Let us suppose that you want to securely send a message *m* from A to B, by guaranteeing ONLY the data confidentiality. For message encryption you should use a symmetric encryption algorithm (since it is faster than asymmetric algorithm). By supposing that A and B share only their public RSA keys KU_A e KU_B (KR_A and KR_B are the private keys), show which functions can be executed at the sender and receiver sides. Try to depict the corresponding schemes.

SOLUTION
Sender:



25) Let us suppose that you want to securely send a message *m* from A to B, by guaranteeing ONLY data authentication/integrity. By supposing that A and B share only a secret key K_{AB} and a hash algorithm H(), show which functions can be executed at the sender and receiver sides. Try to depict the corresponding schemes.

SOLUTION
Sender:



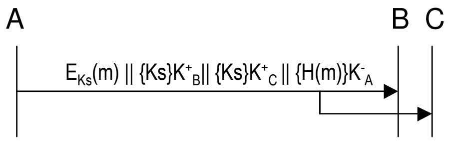
26) Let us suppose that you want to securely send a message *m* from A to B, by guaranteeing both confidentiality and data authentication/integrity. For message encryption you should use a symmetric encryption algorithm (since it is faster than asymmetric algorithm). By supposing that A and B share only their public RSA keys KU_A e KU_B (KR_A and KR_B are the private keys), show which functions can be executed at the sender and receiver sides. Try to depict the corresponding schemes. A and B share the following algorithms: RSA, AES, SHA1.

SOLUTION
Data that are sent:
$x = AES_{Ks}(m) \,||\, RSA_{KUb}(Ks) \,||\, RSA_{KRa}(H(m))$

or also:
$x = AES_{Ks}(m \,||\, RSA_{KRa}(H(m))) \,||\, RSA_{KUb}(Ks)$

27) Let us suppose that you want to securely send a message *m* from A to two recipients B and C, by guaranteeing both confidentiality (through symmetric encryption with algorithm $E_k()$) and data authentication/integrity (through digital signature). Let us suppose that A, B and C have their own private RSA keys, K^-_A, K^-_B e K^-_C , and that they share all their public keys K^+_A, K^+_B e K^+_C.
Please show which functions could be executed by A (sender), and the resulting message *x* that is actually sent from A to B and C.

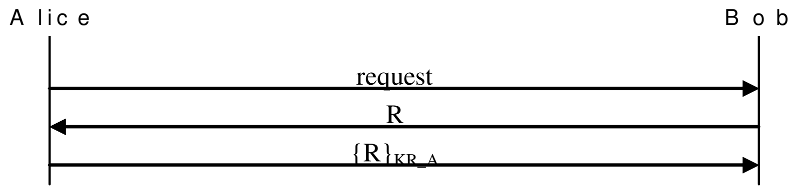
SOLUTION
Sender:



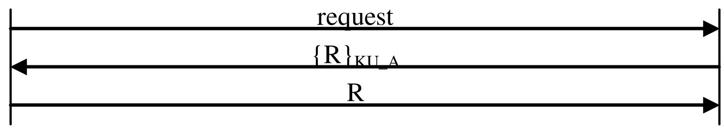
Sent message: $x = E_{Ks}(m) \,||\, \{Ks\}K^+_B \,||\, \{Ks\}K^+_C \,||\, \{H(m)\}K^-_A$

28) Show a possible secure authentication scheme between Alice (supplicant) and Bob (authenticator), by supposing that Alice and Bob share their public RSA keys KU_A and KU_B (KR_A and KR_B are the corresponding private keys).

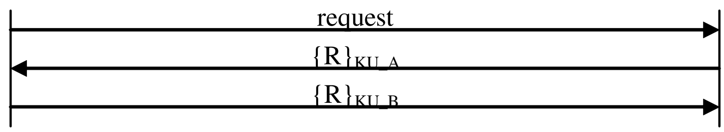
SOLUTION



or:

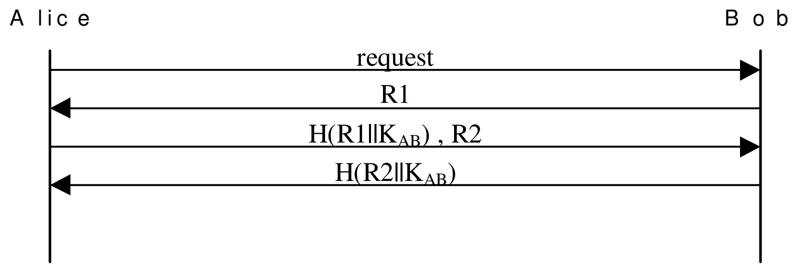


or:



29) Show a possible mutual authentication scheme between Alice and Bob, based on the use of an hash function $H(\cdot)$ and a shared secret K_{AB}.

SOLUTION



30) Consider the following key distribution scheme between A and B through a third party KDC based on symmetric cryptography. The Ka and Kb are the secret keys shared by KDC with A and B, respectively; KS is the new session key:

A	→	KDC:	ID_a, ID_b
KDC	→	A:	ID_b, $\{K_S\}_{Ka}$,$\{Ks\}_{Kb}$
A	→	B:	ID_a, $\{K_S\}_{Kb}$

Show how an intruder C, that is able to intercept and modify the communication between A and B, can attack such a key distribution scheme by letting B believe that he is talking with D (without being able to decrypt the following encrypted communication).

SOLUTION

B has no assurance that the received key *Ks* is actually shared with A (no key authentication). An intruder C that is able to intercept and modify the communication, may force B believe that he is talking with D by changing the message sent by A to B in the following way :

A → KDC: ID_a,ID_b
KDC → A: ID_b,$\{Ks\}K_a$,$\{Ks\}K_b$
A → C: ID_a, $\{Ks\}K_b$
C → B: ID_d, $\{Ks\}K_b$

Note that if C is a valid user of KDC, and if C already received a session key K_{S1} for talking with A (receiving from KDC: ID_a, $\{K_{S1}\}_{Kc}$,$\{K_{S1}\}_{Ka}$),
By changing the message sent by KDC to A (second message) into: ID_b,$\{K_{S1}\}_{Ka}$,$\{K_{S1}\}_{Kc}$
C is able to decrypt flowing messages sent from A to B.

If C already received a session key K_{S2} for talking with B (receiving from KDC: ID_b, $\{K_{S2}\}_{Kc}$,$\{K_{S2}\}_{Kb}$),
by changing the message sent to B (third message) into: ID_a,$\{K_{S2}\}_{Kb}$
C is able to decrypt messages sent from B to A.

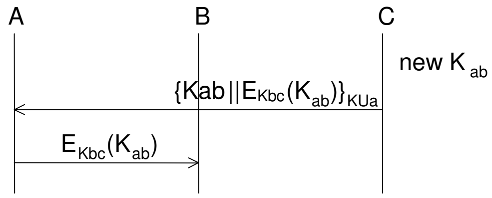
31) Let us consider three entities A, B, and C, and suppose that:

i) A owns a public/private key pair K_A^+/K_A^-;
ii) C has the A's public key, K_A^+;
iii) B and C share a secret key K_{BC};
iv) B and C don't have a direct communication channel;
v) A, B and C do trust each other.

Show a possible message exchange that allows the establishment of a new key K_{AB} for A and B.

SOLUTION

A possible exchange that allows the establishment of a new key K_{AB} for A and B is:



that is:

C → A: $\{ K_{ab}, \{K_{ab}\}_{Kbc} \}_{KA+}$
A → B: $\{K_{ab}\}_{Kbc}$

Note:

In order to protect the key exchange against possible replay attacks, it is possible to use a scheme similar to Needham-Schroeder scheme, where C acts as *trusted-third party* (KDC) between A and B, with the difference that here the public key of A is used between A and C in place of a shared symmetric key K_{ac}. That is:

A → C: ID_a,ID_b,N_a
C → A: $\{Kab,IDb,Na,\{K_{ab},ID_a\}_{Kbc}\}_{KA+}$
A → B: $\{K_{ab},ID_a\}_{Kbc}$
B → A: $\{N_b\}_{Kab}$
A → B: $\{N_b - 1\}_{Kab}$

32) Show a possible key transport scheme between two entities A and B, based on asymmetric encryption (public key cryptography), without the use of a KDC; list the resulting security properties.

SOLUTION

A possible key distribution scheme between A and B is:
$A \rightarrow B$: $\{Ks,sign_A(ID_B,Ks)\}_{KUb}$

This scheme guarantees implicit key authentication to A, key authentication and confirmation to B, but no key confirmation to A. It also doesn't guarantee key freshness to B.

In order to add key freshness guarantee (to B), a timestamp can be also included:
$A \rightarrow B$: $\{Ks,t,sign_A(ID_B,Ks,t)\}_{KUb}$

33) Show an example of authenticated DH exchange that holds out against MITM attack.

SOLUTION
An example of authenticated DH that uses only digital signature is:
$A \rightarrow B$: A, g^{Xa}
$A \leftarrow B$: B, g^{Xb}, $Sign_B(g^{Xa} \| g^{Xb} \| A)$
$A \rightarrow B$: $Sign_A(g^{Xa} \| g^{Xb} \| B)$

An authenticated DH that uses both signature and encryption is (it is a varian of the STS protocol):
$A \rightarrow B$: g^{Xa}
$A \leftarrow B$: g^{Xb}, $E_{Ks}(B \| Sign_B(g^{Xa} \| g^{Xb}))$
$A \rightarrow B$: $E_{Ks}(A \| Sign_A(g^{Xa} \| g^{Xb}))$
Where K_S is a key derived by the DH result g^{XaXb}.

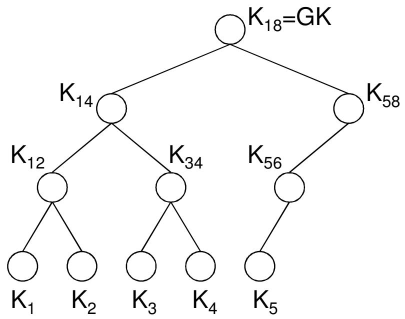
34) Describe a possible message exchange for creating a group key among 4 participants (group members) using a Group Diffie-Hellman key exchange.

SOLUTION
According to GDH, all participants should agree on a prime number p and generator g. Each participant (u_i) generates a secret (private) value x_i.
The finally group key will be $g^{x1x2x3x4}$ $mod\ p$. For computing such key, each participant should receive $y=g^{a\ b\ c}\ mod\ p$ with $a,b,c \neq x_i$ and use x_i for computing $y^{xi}\ mod\ p = g^{a\ b\ c\ xi}\ mod\ p = g^{x1x2x3x4}\ mod\ p$

One possible message exchange could be:
$u1 \rightarrow u2 : g^{x1}$
$u2 \rightarrow u3 : g^{x1x2}, g^{x1}$
$u3 \rightarrow u4 : g^{x1x2x3}, g^{x1x2}, g^{x1}$
$u4 \rightarrow u3 : g^{x1x2x4}, g^{x1x4}, g^{x4}$
$u3 \rightarrow u2 : g^{x1x3x4}, g^{x3x4}$
$u2 \rightarrow u1 : g^{x2x3x4}$

Other message exchanges are possible.

35) Consider the following key tree used by the Logical Key Hierarchy (LKH) group management protocol. Suppose that a new user u6 with secret key K6 needs to be added to the group. Which massages will the Control Center (GKDC) send to all users for updating the keys?



SOLUTION
In order to add the new user u_6 the following key must be changed: K_{18} (the group key), K_{58}, and K_{56}. For sending the new keys to all users, the following messages could be sent to all:
$\{K'_{18}\}_{K14}$, $\{K'_{18}\}_{K'58}$
$\{K'_{58}\}_{K'56}$
$\{K'_{56}\}_{K5}$, $\{K'_{56}\}_{K6}$

Members u_1, u_2, u_3, and u_4 use the shared key K_{14} for obtaining the group key K'_{18}.
Members u_5 and u_6 use their respective keys K_5 and K_5 for obtaining the key K'_{56}. Then they use the key K'_{56} for obtaining key K'_{58}, and use K'_{58} for obtaining the group key K'_{18}.

36) Let us consider an entity A that holds the following digital certificates: $cert_{CA3}(A)$, $cert_{CA2}(CA3)$, $cert_{CA1}(CA2)$, and $cert_{CA1}(CA1)$ (where $cert_Y(X)$ refers to the certificate of X signed by Y). Indicate what A should send to B in order to let A and B start a secure communication, under the following different hypotheses:

SOLUTION

B owns:	A should send to B:
$cert_{CA1}(CA1)$	$cert_{CA3}(A)$, $cert_{CA2}(CA3)$, $cert_{CA1}(CA2)$
$cert_{CA3}(A)$	no additional certificate is required
$cert_{CA1}(CA2)$	$cert_{CA3}(A)$, $cert_{CA2}(CA3)$
$cert_{CA1}(CA1)$, $cert_{CA3}(A)$	no additional certificate is required

37) If A holds $cert_B(A)$ and $cert_C(B)$ (where $cert_Y(X)$ refers to the certificate of X signed by Y), while D holds $cert_E(D)$, please indicate:
   a. what should A hold in order to authenticate D? Show a possible authentication scheme.
   b. what should D hold in order to authenticate A? Show a possible authentication scheme.

SOLUTION
a) what should A hold in order to authenticate D? Show a possible authentication scheme.
The public key of D,
OR the public key of E

In the latter case (A holds the public key of E), a possible authentication scheme is:
D → A: request
A → D: R
D → A: $\{R\}_{KRd}$, $cert_E(D)$
Note: the $cert_E(D)$ can be sent either in the first or in third message.

OR:
D → A: request , $cert_E(D)$

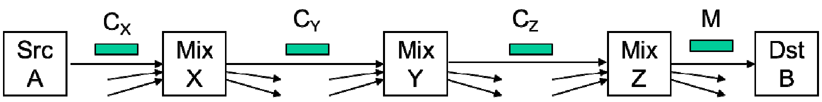
A → D: $\{R\}_{KUd}$
D → A: R

b) what should D hold in order to authenticate A? Show a possible authentication scheme.
The public key of A,
OR the public key of B,
OR the public key of C.

In the latter case (D holds the public key of C), a possible authentication scheme is:
A → D: request
D → A: R
A → D: $\{R\}_{KRa}$, $cert_B(A)$, $cert_C(B)$

OR:
A → D: request, $cert_B(A)$, $cert_C(B)$
D → A: $\{R\}_{KUa}$
A → D: R

38) Let us consider an anonymizing network formed by high-latency anonymizing *Mix* nodes. Let us consider the case in which a node *A* wants to send a message *m* to a node *B* by means of three intermediate *Mix* nodes *X*, *Y*, and *Z*. Assume that K^+_i and K^-_i are respectively the public and private keys of node *i* ($i=x,y,z$).
Indicate the format of the message C_X composed by *A* and sent to the first node *X*.



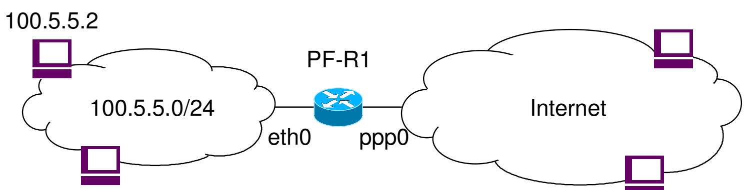
SOLUTION

Data sent by *A* to the first node *X*: $C_X = E_{K^+_x}(ID_Y \| E_{K^+_y}(ID_Z \| E_{K^+_z}(ID_B \| m)))$

where *IDi* is the identify or address of node *i*.

Note:
node *X* will receive such data, decrypt it with K^-_x and relay the following content data to *Y*: $E_{K^+_y}(ID_Z, E_{K^+_z}(ID_B, M))$
node *Y* will receive such data, decrypt it with K^-_y and relay the following content data to *Z*: $E_{K^+_z}(ID_B, M)$
node *Z* will receive such data, decrypt it with K^-_z and relay the message *m* to *B*.

39) Let us consider the following network scheme, where in the node 100.5.5.2 there is a HTTP web server (TCP port 80) and a SMTP mail server (TCP port 25); you are requested to configure the filtering table of the router R1 so that:
   i) from external clients it is possible to access to the internal web server (node 100.5.5.2, TCP port 80);
   ii) from internal clients it is possible to access any external web server (port 80);
   iii) all client/server and server/client communications between the internal SMTP mail server and possible external SMTP servers are enabled; that is, internal SMTP Client → external SMTP Server (TCP port 25), and external SMTP Client → internal SMTP Server (TCP port 25).



SOLUTION

FORWARD								
Matching								action
in_ interface	out_ interface	s_addr	d_addr	Proto	s_port	d_port	state	ACCEPT/ DROP
*	*	*	*	*	*	*	ESTABLISHED	ACCEPT
ppp0	eth0	*	100.5.5.2	TCP	*	80	NEW	ACCEPT
eth0	ppp0	100.5.5.0/24	*	TCP	*	80	NEW	ACCEPT
ppp0	eth0	*	100.5.5.2	TCP	*	25	NEW	ACCEPT
eth0	ppp0	100.5.5.2	*	TCP	*	25	NEW	ACCEPT
*	*	*	*	*	*	*	*	DROP

Or by applying anti-spoofing rules separately:

FORWARD								
Matching								action
in_ interface	out_ interface	s_addr	d_addr	Proto	s_port	d_port	state	ACCEPT/ DROP
ppp0	eth0	100.5.5.0/24	*	*	*	*	*	DROP
*	*	*	*	*	*	*	ESTABLISHED	ACCEPT
*	*	*	100.5.5.2	TCP	*	80	NEW	ACCEPT
*	*	100.5.5.0/24	*	TCP	*	80	NEW	ACCEPT
*	*	*	100.5.5.2	TCP	*	25	NEW	ACCEPT
*	*	100.5.5.2	*	TCP	*	25	NEW	ACCEPT
*	*	*	*	*	*	*	*	DROP

In case of stateless packet-filter, possible workaround that does not use connection state information:

FORWARD								
Matching								action
in_interface	out_interface	s_addr	d_addr	Proto	s_port	d_port	TCP flags	ACCEPT/ DROP
ppp0	eth0	100.5.5.0/24	*	*	*	*	*	DROP
*	*	*	100.5.5.2	TCP	*	80	*	ACCEPT
*	*	100.5.5.2	*	TCP	80	*	*	ACCEPT
*	*	100.5.5.0/24	*	TCP	*	80	*	ACCEPT
*	*	*	100.5.5.0/24	TCP	80	*	SYN=1,ACK=0	DROP
*	*	*	100.5.5.0/24	TCP	80	*	*	ACCEPT
*	*	*	100.5.5.2	TCP	*	25	*	ACCEPT
*	*	100.5.5.2	*	TCP	25	*	*	ACCEPT
*	*	100.5.5.2	*	TCP	*	25	*	ACCEPT
*	*	*	100.5.5.2	TCP	25	*	SYN=1,ACK=0	DROP
*	*	*	100.5.5.2	TCP	25	*	ACCEPT	ACCEPT
*	*	*	*	*	*	*	*	DROP

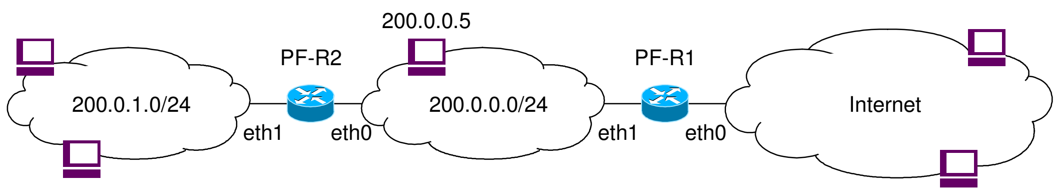
40) Let us consider the following company network formed by an internal network and a DMZ separated by a screening router R2, and connected to the external public network (Internet) through the screening router R1, as shown in figure.
You are requested to configure the filtering table of R1 so that:

a) it is possible to establish application level client→server communications (through any transport protocol) from any DMZ node to any external node;

b) it is blocked any attempt to establish a client→server communication from the external network to the DMZ;

c) it is blocked any communication between the internal and the external networks;

d) it is possible to establish TCP connections from the external network to the node 200.0.0.5 TCP port 80 (HTTP).



SOLUTION

FORWARD								
Matching								action
in_int	out_int	s_addr	d_addr	Proto	s_port	d_port	state	ACCEPT/ DROP
*	*	*	*	*	*	*	ESTABLISHED	ACCEPT
eth1	eth0	200.0.0.0/24	*	*	*	*	NEW	ACCEPT
eth0	eth1	*	200.0.0.5	TCP	*	80	NEW	ACCEPT
*	*	*	*	*	*	*	*	DROP

41) Consider the network of the previous exercise. You are requested to configure the filtering table of R2 so that:

a) it is blocked any attempt to establish a client→server communication from the DMZ to the internal network;

b) it is possible to establish application level client→server communications (through any transport protocol) from any node of the internal network (network address 200.0.1.0/24) to the DMZ;

c) it is blocked any communication between the internal and external network.

SOLUTION

FORWARD								
Matching								action
in_int	out_int	s_addr	d_addr	Proto	s_port	d_port	state	ACCEPT/ DROP
*	*	*	*	*	*	*	ESTABLISHED	ACCEPT
eth1	eth0	200.0.1.0/24	200.0.0.0/24	*	*	*	NEW	ACCEPT
*	*	*	*	*	*	*	*	DROP